Optimizing Encoder for Retrieval via Multi-Vector Late Interaction

Stanford CS224N Custom Project

Xinran Song
Department of Computer Science
Stanford University
xrsong@stanford.edu

Abstract

Neural information retrieval (IR) has greatly advanced search and other knowledge intensive language tasks. Recent research has shown that larger encoders can significantly increase performance of single-vector encoder models such as Google's generalizable T5-based dense Retrievers (GTR). However, the effect of larger encoders have not been tested on ColBERTv2, a leading BERT-based IR system using multi-vector late-interaction mechanism. In this work, **We investigate how the size and pretraining of encoder affect ColBERTv2's in-domain(ID) and out-of-domain(OOD) accuracy.** Compared to the original bert-base encoder, we see 0.4% / 1.2% increase in ID/OOD accuracy and 30% faster indexing with the smaller MiniLM encoder. We also see 0.5% / 1.1% increase in ID/OOD accuracy with bert-large encoder, whereas electra-base shows similar performances to bert-base encoder. Overall, we find MiniLM to be a more optimal encoder model; we also conclude that the encoder pretraining contributes more significantly to model performance than encoder size.

1 Key Information to include

External collaborators (if you have any): None
External mentor (if you have any): Omar Khattab, Keshav Santhanam
Sharing project: None

2 Introduction

Neural information retrieval (IR) has quickly dominated the search landscape over the past 2–3 years, dramatically advancing knowledge-intensive NLP tasks such as document search (Nogueira and Cho, 2019 [1]) and question answering (Guu et al., 2020 [2]). This work builds upon ColBERTv2 [3], a state-of-the-art bi-encoder IR system.

There are two major paradigms in neural retrievers: **single-vector** and **multi-vector**. Single-vector models encode each document and query as single vector, and use their dot-product as the relevance score. On the other hand, multi-vector models like ColBERT encode document and query into multi-vector tokens and uses a rich interaction between the two sets of vectors to compute relevance score. As the interaction happens efficiently at search time, this paradigm is also called *late interaction*. Compared to the single-vector with the final dot-product layer bottleneck, multi-vector architecture is considered more expressive and generalizable.

In a recent paper, Google Research showed that larger encoders can significantly improve out-of-domain(OOD) performance of single-vector models [4]). However, no similar experiments on encoders have been done for multi-vector models like ColBERTv2. Hence, this project aims to answer the following question: *How does varying the size and pretraining of encoder affect the in-domain and out-of-domain (ID/OOD) performance of ColBERTv2?*

The current ColBERTv2 uses the BERT-base-uncased model as encoder for both documents and queries. Therefore, we use bert-base encoder as baseline and experiment with three new models: **MiniLM** [5], **electra-base** [6] and **bert-large** [7]. For each model, we train checkpoints with 100k-200k steps, then compare their index time and ID/OOD accuracy with the baseline. We observe 1% increase in ID/OOD accuracy for both MiniLM and bert-large models, despite MiniLM being smaller and 3x faster in indexing than bert-large.

This work makes the following contributions:

1. We discover a better MiniLM-based encoder with 0.6x index time and 1.2% higher OOD performance than the current bert-base encoder.
2. We find that encoder pretraining has larger effect on retrieval accuracy than encoder size.

3 Related Work

In recent years, there have been extensive research and new approaches on both single-vector and multi-vector retrieval models.

Traditional single-vector models include Dense Passage Retrieval (DPR) introduced in Karpukhin et al.[8], which uses BERT to encode passages and queries into single dense vectors, with dot-product similarity as relevance score.

For multi-vector models, a paper by Khattab & Zaharia in 2020 [9] proposed the multi-vector late-interaction based ColBERT system, encoding documents and queries into sets of token vectors. In addition to ColBERT, the SPLADE system [10] also performs token-level late interaction, but reduces tokens into one-dimension.

A subsequent work by Khattab et al. in 2022 [3] presents a state-of-the-art COlBERTv2 model optimized from ColBERT. One key method of optimization is supervision with distilled tuples in training. Instead of standard (query, positive doc, negative doc) triples, ColBERTv2 is fine-tuned with n-way tuples (query, 1 positive doc, $n-1$ lower-ranked docs) with scores from a cross-encoder reranker. We adopt the ColBERTv2 model as baseline and fine-tune different encoders with the same n-way tuples.

Soon after ColBERTv2, work from Google Research [4] showed that larger encoders enables dual encoder system (the GTR system) to overcome the dot-product bottleneck and generalize even better than models like ColBERTv2. The Ni et al. paper shows that retrieval performances steadily improves across model sizes from GTR-base (110M params) to GTR-XXL (4B params). This inspires us to investigate whether the same pattern holds for ColBERTv2's late-interaction based architecture.

4 Approach

Overall, this project builds upon the existing ColBERTv2 architecture, and compares the retrieval performance (both ID & OOD) and indexing time of 4 different encoders.

4.1 Baseline The current ColBERTv2 encoder is fine-tuned from BERT-base-uncased model (110M params) with token vector dimension of 128 and 64-way distillation tuples. In this paper, for faster iterations, we use as baseline the **bert-base-uncased** model with token vector dimension of **128**, fine-tuned with **8-way** distillation tuples. See Section 5 for index time and ID/OOD accuracy of the baseline model.

4.2 Encoder Model Selection We picked 3 models across different sizes and pretraining methodologies.

For the large model, we use the bert-large-uncased from HuggingFace [11], with the same pretraining as our baseline bert-base but scaled up to 336M parameters.

We also include a model of the same size as bert-base but different pretraining: electra-base-discriminator from HuggingFace. According to the original paper [6], Electra's pretraining with replaced token detection enables it to outperform models of same size and data. We shall see if the improvements apply to ColBERTv2 as well.

For the small model, we use microsoft/MiniLM-L12-H384-uncased from HuggingFace [5]. The MiniLM model has 33M parameters and is distilled from the UniLMv2 model [12]. This is a combination of changes in both size and pretraining compared to our baseline model bert-base.

4.3 Original Contribution in Code Based on the existing ColBERTv2 codebase [13], we implemented custom pipeline of scripts for fine-tuning encoders, indexing documents, measuring indexing time, searching/ranking documents and evaluating ID/OOD accuracy. We also had to modify the existing codebase to enable training with multiple models as encoder (see code upload).

5 Experiments

5.1 Training Data Based on the existing ColBERTv2 codebase[13], we finetune each model with tuples generated through cross-encoder. We first used the **MS MARCO Passage Ranking Train Triples** [14](query, positive passage, negative passage) to train a ColBERTv1 model. Then, we used the ColBERTv1 model to rank top-k passages and pass them through a cross-encoder to generate n-way ($n = 8$) tuples with relevance scores. These tuples are used to fine-tune models for ColBERTv2.

5.2 Evaluation method For in-domain (ID) model evaluation, we use the **MS MARCO Passage Ranking Top 1000 Dev Set** [14]. The evaluation metric used is one of MS MARCO's default metrics, MRR@10: $MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$.

For out-of-domain (OOD) evaluation, we use the **LoTTE Dev Set** [3]. The dataset contains 12 topic-stratified test sets, each with 500–2000 queries and 100k–2M passages across 5 topics (Writing, Recreation, Science, Technology, Lifestyle), as well as an aggregated Pooled setting. We use LoTTE's built-in Success@5 for the Pooled dataset as evaluation metric, where a point is awarded to the system for each query where it finds an accepted answer from the target page in the top-5 hits.

For both ID and OOD evaluation, we choose to use the test sets' default metrics (MRR@10 for MS MARCO, Success@5 for LoTTE) to facilitate comparisons with other systems in existing literature. To examine the model's ID and OOD improvement together, we also calculate
OOD-to-ID ratio = OOD-improvement / ID-improvement. A large ratio shows that the model improves OOD generalization more significantly.

5.3 Experimental details We run training (for checkpoint steps of 100-200k), indexing, and evaluation on four models: **bert-base (baseline)**, **MiniLM** [5], **electra-base** [6] and **bert-large** [7]. Bert-base, bert-large, amd MiniLM are trained with learning rate of 1e-05, embedding dimension of 128 and distilled tuples with 8 negatives. Due to time constraint, we reduced the dimension to 64 for comparison between bert-base and electra-base.

For each model, we index the LoTTE Pooled corpus (2.8M passages) with 4 Titan V GPUs in a multi-core machine with 56 CPU cores. We record the index time in order to compare time efficiency between models of different sizes.

5.4 Results For each of the 3 models (bert-large, electra-base, miniLM), we report accuracy comparison between the model and baseline (bert-base) for finetuning steps 100k, 150k and 200k in two plots, one for ID evaluation and one for OOD evaluation.

In addition to model-specific results, we report a table with index time for each model. We also present a summary table of the best scores of each model in MS MARCO MRR@10 and LoTTE Success@5, in comparison to baseline and other systems in existing literature.

1. **Bert-large vs. Bert-base**
   In Figure 1 and 2, we see that bert-large generally performs better than bert-base. Between checkpoint steps 100k and 200k, bert-large achieves best MRR@10 of 38.8 for MS MARCO, **0.5** points higher than the best MRR@10 for bert-base. For OOD performance, best Success@5 for bert-large beats bert-base by **1.1** points. The OOD-to-ID ratio (as defined in Section 5.2) is **2.2**. It is also worth noticing that bert-large converges to best performance at 100k, earlier than the smaller bert-base model.

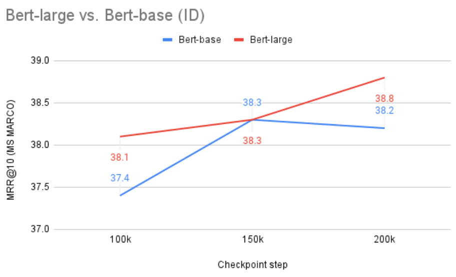

Figure 1: Bert-large ID Comparison

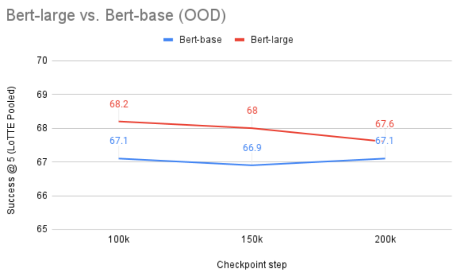

Figure 2: Bert-large OOD Comparison

2. **Electra-base vs. Bert-base**

   Figure 3 and 4 shows the ID/OOD performance of electra-base compared to bert-base. Due to time constraint, we compare electra-base to bert-base with dim=64 (instead of dim=128), we also did not evaluate the models at the 150k step checkpoint. Interestingly, we see that electra-base does better than bert-base on MS MARCO (by **0.2** points in best MRR@10), but with worse out-of-domain performance in LoTTE (worse by **0.1** point in Success@5). Hence, the OOD-to-ID ratio is **-0.5**, which is the only negative ratio among the 3 models. We will further analyze electra-base's OOD performance by looking into category breakdowns of LoTTE in Section 6.1.

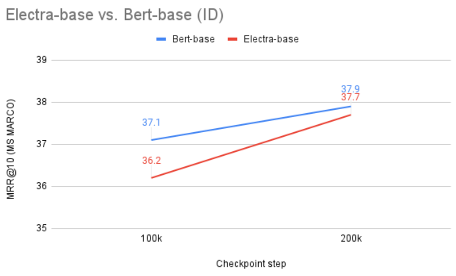

Figure 3: Bert-large ID Comparison

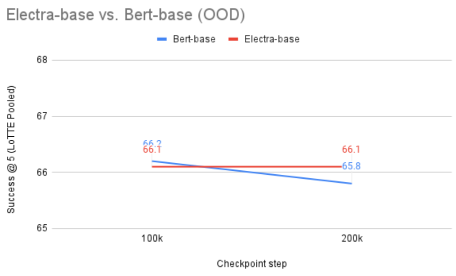

Figure 4: Bert-large OOD Comparison

3. **MiniLM vs. Bert-base**

   Figure 5 and 6 shows the comparison between MiniLM and baseline. MiniLM performs reasonably better than bert-base in MS MARCO (with **0.4** point improvement), but with much larger OOD improvement of **1.2** point, leading to the highest OOD-to-ID ratio of **3** among all three models in our experiments. Overall, MiniLM shows the most pronounced improvement in out-of-domain generalization relative to in-domain improvements. We will analyze the possible causes for this success in Section 6.2.

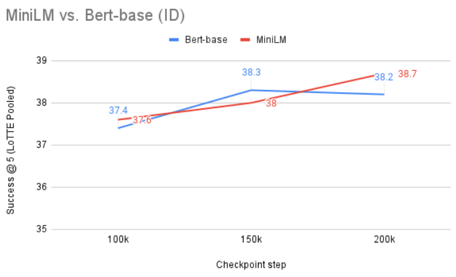

Figure 5: Bert-large ID Comparison

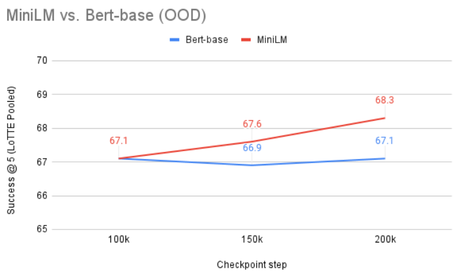

Figure 6: Bert-large OOD Comparison

4. **Indexing time of each model**

   We observe a definite trend of increasing index time with larger models. Combared to baseline bert-base, bert-large's index time is over 2x, and MiniLM's index time is 75% that of bert-base. See Table 1.

	bert-base	electra-base	bert-large	miniLM
# of params	110M	110M	336M	33M
Index Time	2743s	2473s	5109s	1817s

Table 1: Size and Index Time of Models

5. **Table of Best Scores**

   Table 2 compares the best evaluation results of Bert-large and MiniLM to our baseline (bert-base), ColBERTv2, and a few other models from literature (we got results of other models from the original ColBERTv2 paper [3]). We sample a few models both with and without distillation or special pretraining. (For consistency, we omit results from electra-base because it is trained with dim=64 due to time constraint, all other ColBERTv2 models have dim=128).

	MRR@10 (MS MARCO)	Success@5 (LoTTE Pooled)
Models without Distillation or Special Pretraining		
DPR	31.1	-
BM25	-	48.3
ANCE	33.0	66.4
Models with Distillation or Special Pretraining		
SPLADEv2	36.8	68.9
RocketQAv2	38.8	69.8
ColBERTv2	39.7	71.6
Models in this work (with Distillation)		
Bert-base	38.3	67.1
Bert-large	38.8	68.2
MiniLM	38.7	68.3

Table 2: Best ID/OOD Results from Models

   From Table 2, we can see that the models in this project have significantly better results than the non-distilled models, which showcases the strength of the distilled n-way tuples in fine-tuning. All three models in this work performs worse than ColBERTv2, which is expected as ColBERTv2 is fine-tuned with 64-way tuples instead of 8-way tuples in this work.

   Despite being fine-tuned with 8-way tuples only, Bert-large and MiniLM still achieves comparable ID / OOD performance as SPLADEv2 and RocketQAv2 (with delta smaller than 1%).

   Among the new models, MiniLM and bert-large performs almost equally well, achieving 0.5% increase in MS MARCO MRR@10 and 1.2% increase in LoTTE Success@5 compared to baseline. It is reasonable to believe that if we train either Bert-large or MiniLM encoder with the full-fledged configurations (64=way tuples etc), it would surpass the performance of current ColBERTv2. Moreover, MiniLM encoder would enable both higher performance and shorter index time.

6 Analysis

In this section, we dig deeper into two experiment results. First, why electra-base performs slightly worse OOD than bert-base (and much worse than the other 2 models); second, why MiniLM performs equally well as bert-large despite being smaller and almost 3x faster to index than bert-large.

6.1 Topic-wise OOD Analysis First, since the LoTTE dataset arranges passages into five topics (subgroups of the Pooled dataset from which we measured OOD Success@5), we first investigate the models' topic-wise performance. Figure 7 shows each model's 'delta' with respect to baseline at their best performance checkpoint step (100k for electra-base and miniLM, 200k for bert-large), positive delta signifies improvement and vice versa.

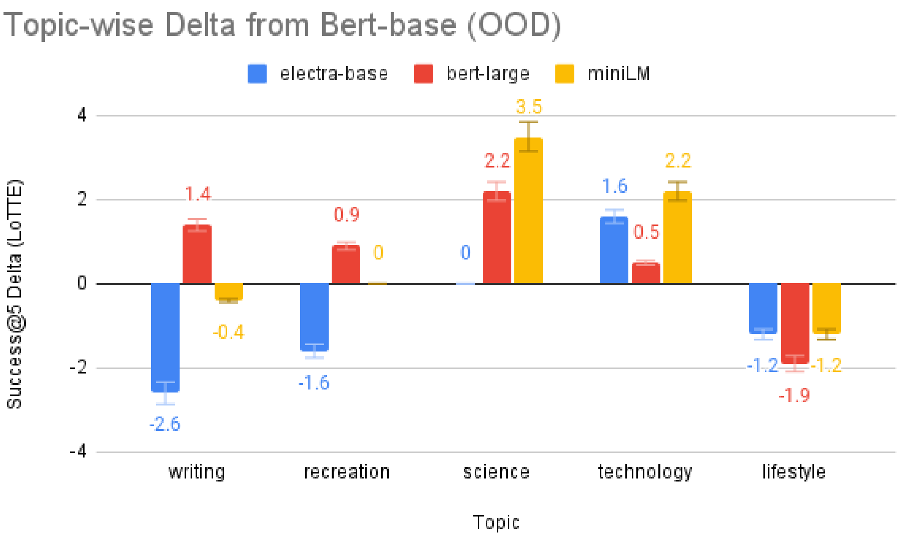

Figure 7: Topic-wise OOD Delta from Bert-base

From Figure 7, we can see three trends. First, electra-base mostly loses out on *writing*, *recreation* and *science* compared to the other two models. Second, MiniLM does exceptionally well on *science* and *technology*. Three, all three models performs worse than bert-base in *lifestyle*.

The exact reasons for these observations require further investigations in future work, here we propose a few hypotheses. First, the semantics of queries/documents from different topics might differ, and certain pretraining methods (replaced-token-detection for electra, distillation for miniLM, etc) may be more/less suited to a particular semantic pattern. Second, of all topics, *lifestyle* is the most general and encompasses the widest range of sub-topics, this may contribute to its being particularly hard for models to improve on.

6.2 Reason for MiniLM's Success As shown by the above experiments, MiniLM combines the best of both worlds: faster indexing and better ID/OOD performance.

This aligns with the observation in the original MiniLM paper [5] that "(the model) retains more than 99% accuracy on SQuAD 2.0 and several GLUE benchmark tasks using 50% of the Transformer parameters". This is largely thanks to the deep self-attention distillation which enables the model to preserve most of the teacher model's power with fewer parameters.

In our case, as described in Section 4.2, the model is distilled from UniLMv2. According to the original literature [12], UniLMv2 outperforms BERT in multiple benchmarks, which might also explain MiniLM performs equally well (even slightly better) as Bert-large despite its small size.

7 Conclusion

7.1 Summary First, among the three models, we find that MiniLM to be the optimal encoder with improved performance (increase by 0.4 point ID and 1.2 point OOD) and only 0.6x index time as the current baseline encoder. We find that bert-large shows similar improvements as MiniLM but 2x index time as the baseline. Electra-base does not show any significant improvement.

Second, examining model pretraining, size, and performance, we conclude that model pretraining contributes more to performance than model size. Especially in the case of MiniLM, where distillation in pretraining allows the model to retain most capacities with much fewer parameters, we confirm that this pattern continues to hold in systems like ColBERTv2.

7.2 Limitations Due to time constraint, we were obliged to run experiments on electra-base model with reduced dimension of 64 instead of 128. We also did not run experiment on the 150k checkpoint step. With more time, these experiments would put electra-base to a more similar settings as the other 2 models.

Moreover, as mentioned in Section 6.1, further analysis could be done on topic-wise OOD analysis to better explain the varying performance of certain models/topics. For instance, as a future extension we might sample common 'misses' in each topic to look for shared features.

7.3 Future Work Looking back at Table 2, we see a significant gap between models in this work and the ColBERTv2 benchmark reported by the original paper. The major difference lies in fine-tuning data, where 64-way distilled tuples are reduced to 8-way for faster iterations in this work. The large gap in performance (1.0 point for ID, 4.5 points for OOD) reflects the importance of fine-tuning with distilled tuples. One interesting direction of future exploration could be modifying the fine-tuning methodology (n-way of tuples, new ways of generating tuples, etc).

References

[1] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2019.

[2] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training, 2020.

[3] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction, 2021.

[4] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021.

[5] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

[6] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[8] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering, 2020.

[9] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery.

[10] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking, 2021.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[12] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training, 2020.

[13] https://arxiv.org/abs/2104.08663.

[14] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.